

```

*****
*                                     Copyright 2008                               *
* NOMBRE DEL ARCHIVO: JC_LAB_3.asm                                              *
*                                     *                                              *
* PROPÓSITO: Mostrar turnos desde 00 hasta 99                                *
*               utilizando un interruptor para asignar turno                    *
*                                     *                                              *
* DISPOSITIVO: MC68HC908GP32CP                                                *
*                                     *                                              *
* MEMORIA UTILIZADA:                                                           *
*               RAM:          ROM:                                             *
*                                     *                                              *
* ENSAMBLADOR: CASM08-68HC08          VERSION: 2.3                            *
*                                     *                                              *
* DESCRIPCIÓN: Este programa incrementa un contador, lo                      *
*               despliega en doble display de 7 segmentos,                   *
*               utiliza decodificador 7447                                     *
* AUTOR: PE          LOCACIÓN: PEI FECHA INI: 2008/05/08                       *
*               FECHA FIN: 2008/XX/XX                                         *
*                                     *                                              *
* HISTORIA DE ACTUALIZACIÓN                                                    *
* REV      AUTOR      FECHA      DESCRIPCIÓN DE CAMBIOS                      *
* ---      - - - - -      - - - - -      - - - - -                          *
*                                     *                                              *
*=====*
*****
$Include 'gpregs.inc'

*****
* Definición de constantes
*****
RESET_VEC      equ      $FFDC          ; Los precesadores Motorola HC08 poseen un bus de
                                         ; datos de 16 bits y adicionalmente son BIGENDIAN
FLASH_START    equ      $8000          ; Inicio de la memoria FLASH del GP32
RAM_START       equ      $0040

*****
* Definición de variables
*****
                org      RAM_START          ; Aqui se definen variables

unidad         rmb      1
decena         rmb      1
copd           equ      0

*****
* Inicio del programa

```

INICIO org FLASH_START

```

bset    0,CONFIG1    ; Deshabilita el COP del GP32
rsp      ; Inicializa el Stack Pointer(SP) en la parte alta de
           ; página 0
clr      ; inicializa el registro A en cero
clrx     ; inicializa el registro X en cero
clr      ; inicializa el puerto A en cero
portb    ; inicializa el puerto B en cero
portc    ; inicializa el puerto c en cero
unidad   ; inicializa variables en cero
decena   ; inicializa variables en cero

mov      #$03,ddra    ; Pone 2 pines del puerto A como salida
mov      #$ff,ddrb    ; Pone 8 pines del puerto B como salida
mov      #$00,ddrc    ; Pone 8 pines del puerto c como entrada

```

multiplexar:

```

ldx      unidad      ; Parte baja (X) del registro INDEX cargada con valor de "unidad"
LDA      TABLA,X     ; Carga "acc" con valor de la "tabla" que corresponde al valor de reg x
STA      PTB         ; Carga el valor de Acc en puerto B

bclr     0,porta      ; pone en BAJO el bit PTA 0 (Multiplexado es activo bajo) ("unidad" on )
bset     1,porta      ; Pone en ALTO el bit PTA 1 ("decena" off)
jsr      retardo      ; Ejecuta la rutina de retardo

bset     0,porta      ; Pone en ALTO el bit PTA 0 ---puede quitarse-- ("decena" off)
bset     1,porta      ; Pone en ALTO el bit PTA 1 ("unidad" off)

ldx      decena      ; Parte baja (X) del registro INDEX cargada con valor de "decena"
LDA      TABLA,X     ; Carga "acc" con valor de la "tabla" que corresponde al valor de reg x
STA      PTB         ; Carga el valor de Acc en puerto B

bclr     1,porta      ; Pone en BAJO el bit PTA 1 (Multiplexado es activo bajo) ("decena" on )
bset     0,porta      ; pone en ALTO el bit PTA 0 ---puede quitarse-- ("unidad" off)
jsr      retardo      ; Ejecuta la rutina de retardo

bset     0,porta      ; Pone en ALTO el bit PTA 0 ("decena" off)
bset     1,porta      ; Pone en ALTO el bit PTA 1 ("unidad" off)

brclr    0,portc,incrementar_u ; Salta a "incrementar_u:" si el PTC 0 es cero (pulsador presionado)
bra      multiplexar  ; Salta a "multiplexar:"

```

limpia_unidad:

```

JC_LAB_3_V_23.txt
clr          ; Pone en CEROS parte baja del INDEX
clr          ; Pone en CEROS el valor de "unidad"
inc          ; Incrementa el valor de "decena"
lda          ; Carga "ACC" con valor de "decena"
cbeqa       #$0A,limpia_decena
bra          ; ----> PENDIENTE PENDIENTE PENDIENTE <----
             aqui
limpia_decena:
clr          ; Pone en CEROS parte baja del INDEX
clr          ; Pone en CEROS el valor de "decena"
clr          ; Pone en CEROS el valor de "unidad"

incrementar_u:
inc          ; Incrementa el valor de "unidad"
lda          ; Carga el ACC con el valor de "unidad"+
cbeqa       #$0A,limpia_unidad
             ; ----> PENDIENTE PENDIENTE PENDIENTE <----

aqui:
brclr       0,portc,aqui
bra         multiplexar
             ; Salta a "multiplexar:"

retardo:
             ; TOTAL CICLOS 07, RELOJ 10 Mhz, Periodo x ciclo = 10 MICROSEGUNDOS
             ; Veces que se ejecuta 5
             ; retardo 7 ciclos x 5 repeticiones x 10 microsegundos = 350

microsegundos
lda         #$05
dbnza      *
rts         ; Carga "Acc" con valor [A6 05 --> ciclos 02]
             ; Decrementa "Acc" y sale del loop si "Acc" es "Cero" [4B--> ciclos 03]
             ; Retorna a la dirección siguiente de la que fue invocada (¿ PC<--PC+1 ?)
             ;[81 --> ciclos 04]

TABLA:
             ; Tabla utilizando decodificador 7447 (7 segmentos anodo comun)
DB          $00
DB          $01
DB          $02
DB          $03
DB          $04
DB          $05
DB          $06
DB          $07
DB          $08
DB          $09
             ; Numero 0
             ; Numero 1
             ; Numero 2
             ; Numero 3
             ; Numero 4
             ; Numero 5
             ; Numero 6
             ; Numero 7
             ; Numero 8
             ; Numero 9

```

```
*****
* Para el cálculo del retardo se empleo
* la siguiente fórmula:
*  $((8 \times HX + 3) \times A + 2) = \text{\#ciclos}$ 
* donde  $\text{ciclos} = 0.5 / (1 / (\text{Frecuencia\_bus})) = 614400$ 
* suponiendo un XTAL de 4.9152MHz
* Con A(Parámetro) = 5 ---> HX = 15359.2
*****
```

```
dummy_isr:
```

```
    RTI                ; regreso
```

```
*****
```

```
* Vectores
```

```
*****
```

```
                org      RESET_VEC        ; Posicionar el ensamblador
                                                ; en la dirección Reset_vec

    dw  dummy_isr      ; Vector base de tiempo
    dw  dummy_isr      ; Conversion ADC Completa
    dw  dummy_isr      ; Vector de teclado
    dw  dummy_isr      ; Vector de transmision SCI
    dw  dummy_isr      ; Vector de recepcion SCI
    dw  dummy_isr      ; Vector de error SCI
    dw  dummy_isr      ; Vector de transmision SPI
    dw  dummy_isr      ; Vector de recepcion SPI
    dw  dummy_isr      ; Vector de sobreflujo TIM2
    dw  dummy_isr      ; Vector canal 1 TIM2
    dw  dummy_isr      ; Vector canal 0 TIM2
    dw  dummy_isr      ; Vector de sobreflujo TIM1
    dw  dummy_isr      ; Vector canal 1 TIM1
    dw  dummy_isr      ; Vector canal 0 TIM1
    dw  dummy_isr      ; Vector PLL
    dw  dummy_isr      ; Vector ~IRQ1
    dw  dummy_isr      ; Vector SWI
    dw  INICIO         ; Definir en Reset_vec la dirección de Inicio.
```

```
*****
```

```
* preguntas < correojuanca@gmail.com >
```

```
*
```

```
*****
```